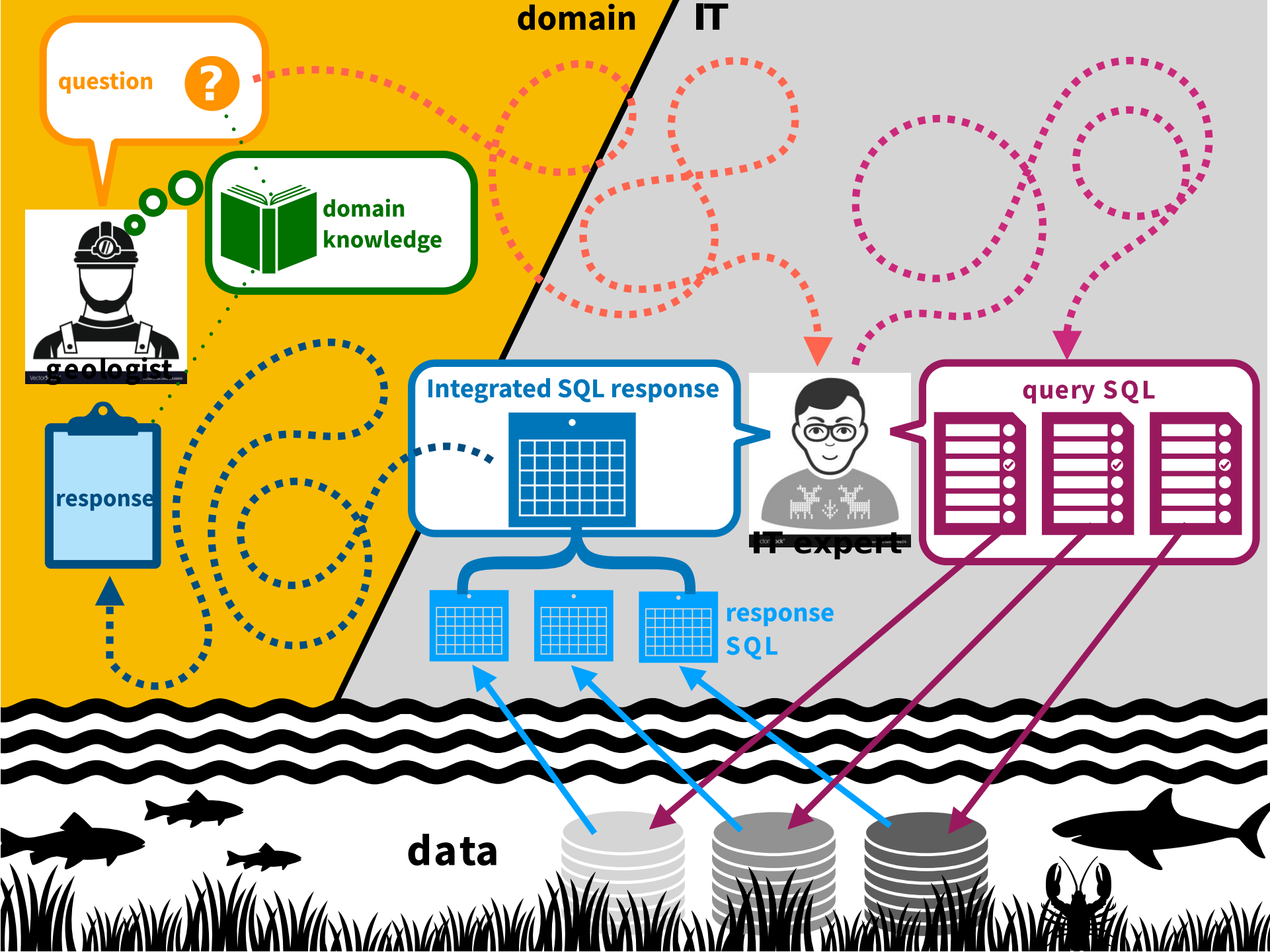


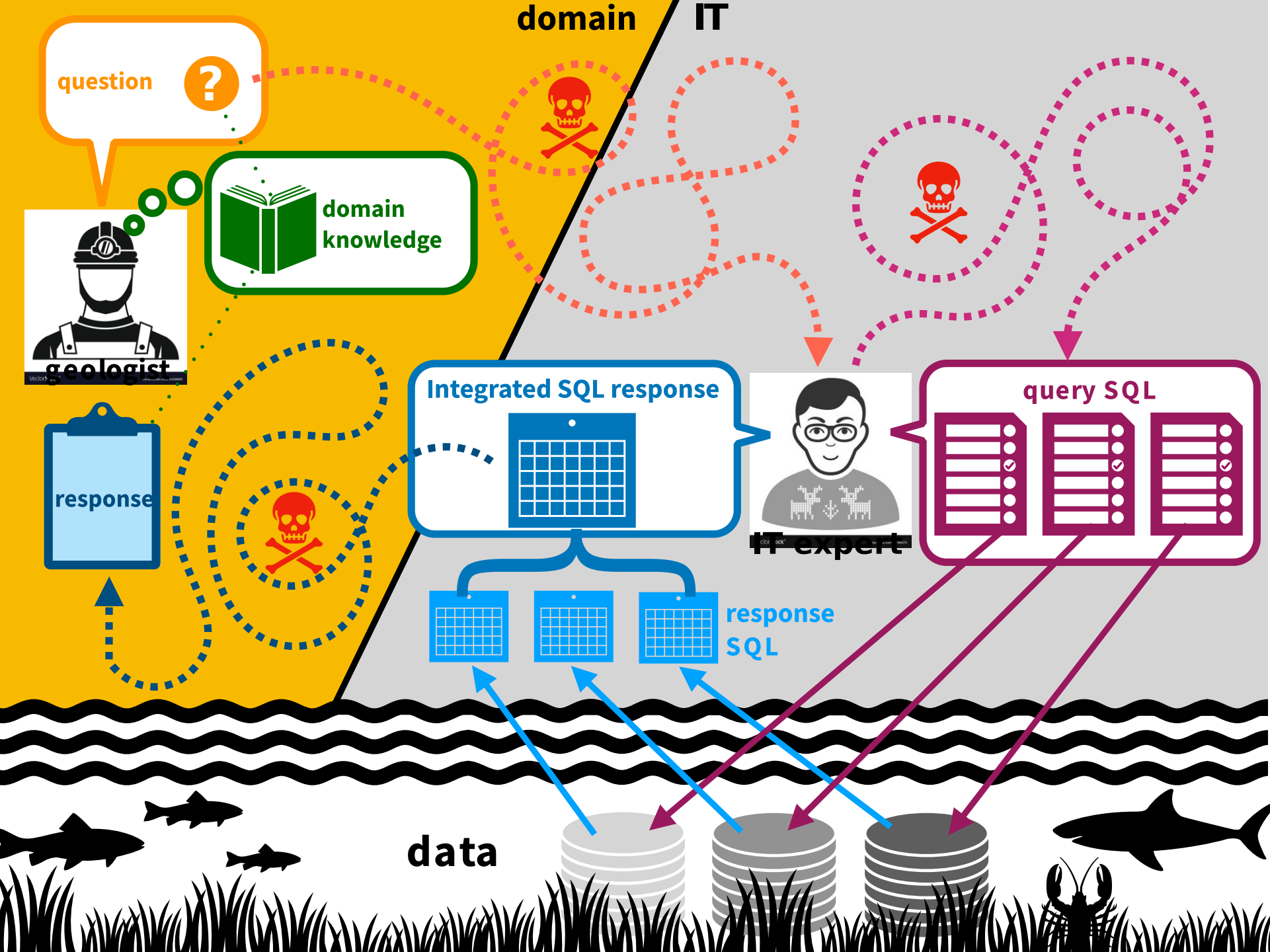
# **Ontopic**

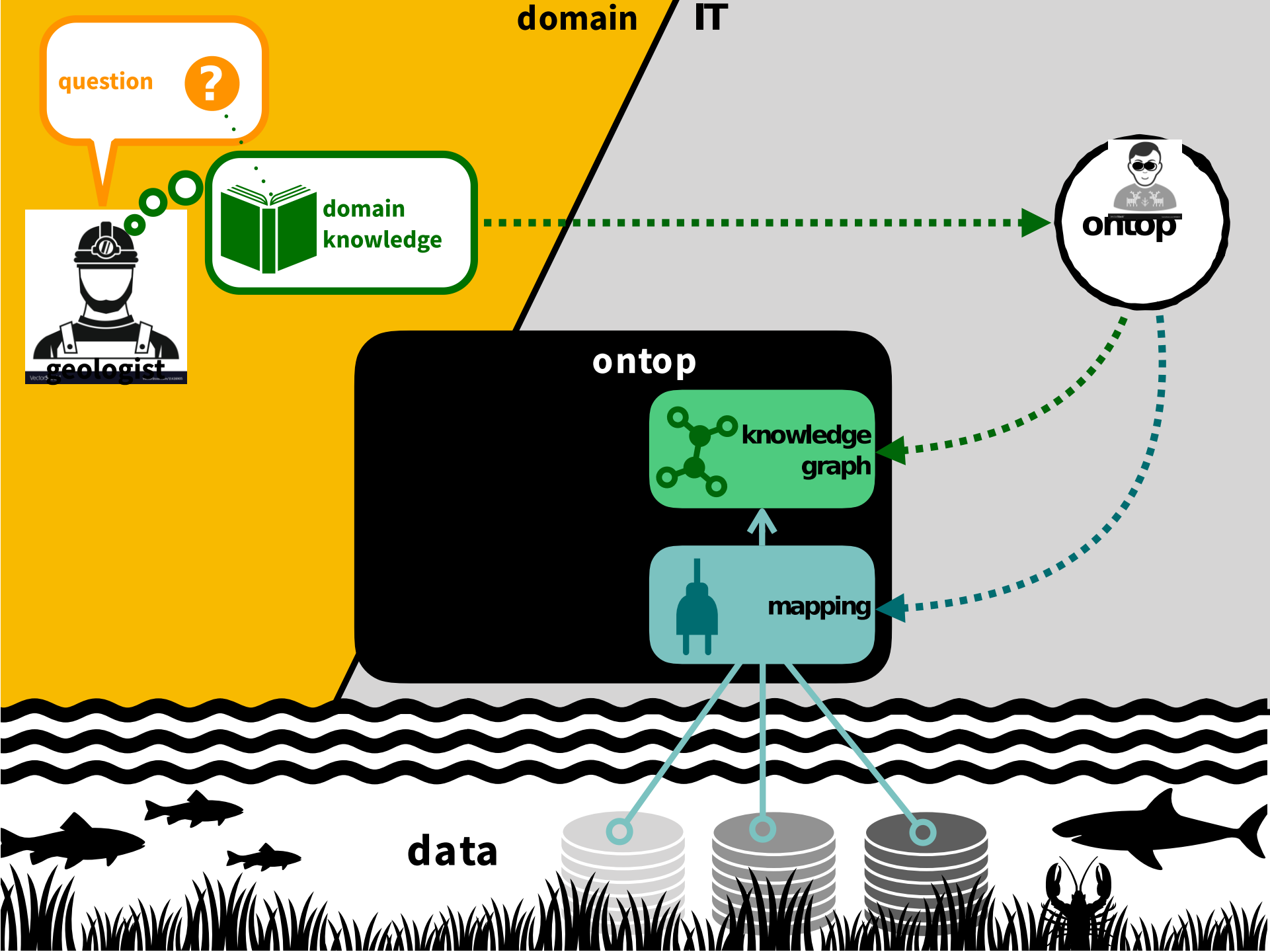
## **The Virtual Knowledge Graph Company**

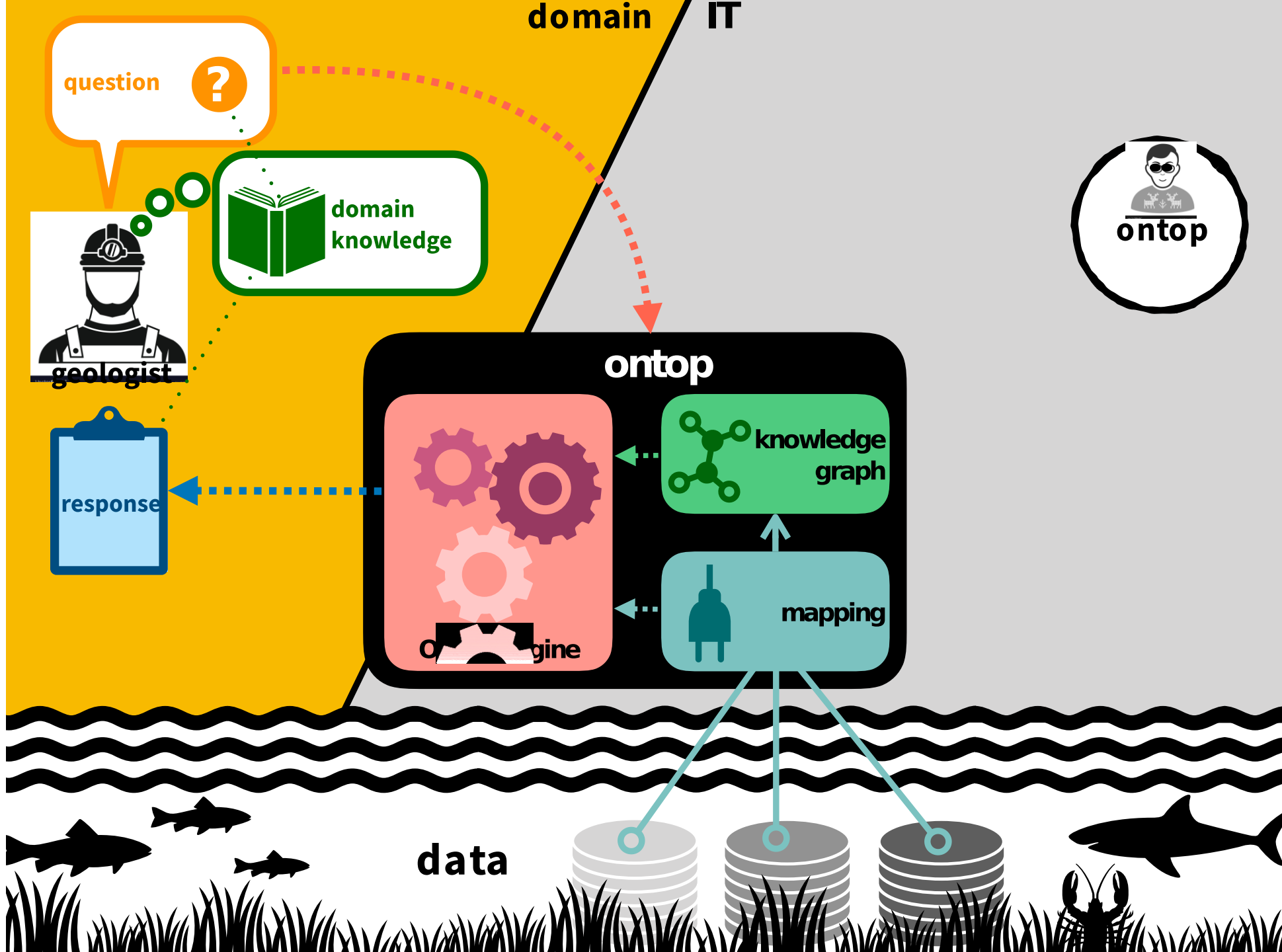
Peter Hopfgartner  
Benjamin Cogrel

Ontopic GmbH









# Virtual Knowledge Graphs (a.k.a. OBDA)

- VKGs: A novel way of accessing existing data
- Knowledge Graph:
  - flexible data integration
  - exploit domain/business knowledge
  - reasoning without performance penalty
- Virtual: data stays in the data sources
  - non-intrusive
  - low initial investment

# VKGs for Data Integration

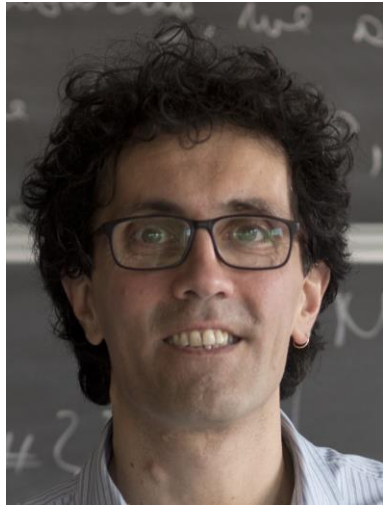
- Data integration made tractable
  - Agile
  - easily maintainable
  - costs are kept low
- Result: data ready for business users
  - intuitive domain vocabulary
  - users don't have to care about data sources
  - different perspectives on the data
  - allows for answering a variety of questions

# Ontop: The Engine for VKGs

- Advanced VKG engine
  - supports answering complex queries over VKGs
  - relies on availability of ontology and mapping to data sources
- Compliant with main W3C standards and major DBMS engines
- Development
  - builds on solid formal foundations (award winning publications)
  - carried out in the context of EU projects
  - open source (Apache 2 licence) and community-based
  - 2 co-founders are among the core developers
  - already widely adopted, including industry and services



# The Team: Roles in Ontopic



**Diego  
Calvanese**  
Scientific  
Advisor



**Benjamin  
Cogrel**  
CTO



**Guohui Xiao**  
Chief  
Scientist



**Peter  
Hopfgartner**  
CEO



**Marco  
Montali**  
Scientific  
Consultant

# Products and Services

- Technical services
  - Technical support for Ontop
  - Specific developments
- Consulting
  - Integration of VKG in existing IT landscapes
  - Integration of Ontop in specific products

**Ontop demo**

# Thank you for attending

Still some questions?

Ontop is on GitHub

<https://github.com/ontop/ontop>



### Edit Mapping

Mapping ID:

Target (Triples Template):  
<http://musicbrainz.org/artist/{gid}#\_> a :SoloMusicArtist .

Source (SQL Query):  
SELECT \*  
FROM artist  
WHERE artist.type = 1

(100 rows)

### Edit Mapping

Mapping ID:

Target (Triples Template):  
<http://musicbrainz.org/artist/{gid}#\_> :member\_of  
<http://musicbrainz.org/artist/{band}#\_> .

Source (SQL Query):  
SELECT a1.gid, a2.gid AS band  
FROM artist a1  
INNER JOIN l\_artist\_artist ON a1.id = l\_artist\_artist.entity0  
INNER JOIN link ON l\_artist\_artist.link = link.id  
INNER JOIN link\_type ON link\_type = link\_type.id  
INNER JOIN artist a2 on l\_artist\_artist.entity1 = a2.id  
WHERE link\_type.gid='5be4c609-9afa-4ea0-910b-12ffb71e3821'  
AND link.ended=FALSE

(100 rows)